Lectures on November 6th & 8th

Prepared by : *Dantong Ji, Xiaodi Wu*

Quantum Interactive Proofs and QIP = PSPACE

1 Introduction

We have discussed the complexity class IP in class. We know from before that IP = PSPACE, showing the vast computational power of the interactive proof model. What about introducing some quantum flavor into the interactive proof system? If both the prover and the verifier can do quantum computation, and they are allowed to communicate with quantum information, we get a new complexity class QIP as the quantum analog of IP. Clearly IP $\subseteq$ QIP, since we can simulate classical computation, thus PSPACE = IP $\subseteq$ QIP. A natural question is whether we can gain extra computational power with quantum setting, namely whether PSPACE $\subsetneq$ QIP. Unfortunately the answer is no. It is proved that QIP $\subseteq$ PSPACE, as we are going to show in this lecture note.

This striking result is first proved by Jain, Ji, Upadhyay and Watrous in 2009[1], formulating as a semidefinite programming problem. Then it is simplified by Wu in 2010[2], establishing a connection between a QIP-complete problem and the computation of some equilibrium value.

We start from a QIP-complete problem *close images*, and convert it into a min-max problem, which can be solved by the *matrix multiplicative weights update method*. Since this method can be implemented using a polynomial-space uniform family of Boolean circuits having polynomial-depth, namely *close images* $\in$ NC(poly), together with the fact that NC(poly)=PSAPCE[3], we prove that QIP $\subseteq$ PSPACE. The whole lecture note will roughly follow these steps.

Readers are assumed to have basic acquaintance with quantum computation, though we do give a brief explanation of quantum proof systems. This lecture note will go through the whole proof that QIP = PSPACE. And as stated above, the major difficulty is to prove QIP $\subseteq$ PSPACE. Even though we want to be as self-contained as possible, we cannot dig into every detail due to the limited space. In this lecture note, we mainly focus on the conversion to a min-max problem, and the matrix multiplicative weights update method adopted in this case. In terms of the proofs to some conclusions mentioned, we will point the readers to related materials when appropriate.

Be aware that even if QIP = IP, it does not mean quantum computation earn us nothing over classical interactive proof systems. In the very brief summary of the quantum interactive proof system coming in the next section, we will present some known results useful in our later proof, which is both elegant and surprising.

2 Preliminaries

2.1 Basic Quantum Notation

A quantum system with k qubits is call a register X, and mathematically taken as a Hilbert space $\mathcal{X}$ with dimension of 2^k. The computational basis is an orthonormal basis of great importance, which is often written using Dirac notation as $\{|x\rangle : x \in \Sigma^k\}$, where $\Sigma = \{0, 1\}$. All pure states can be expressed using a column vector $|\psi\rangle = \sum_{x\in\Sigma^n} \alpha_x |x\rangle$, with $\sum_x \alpha_x^2 = 1$. Its conjugate transpose is written as $\langle\psi|$. The tensor product of two Hilbert spaces $\mathcal{X}, \mathcal{Y}$ is written as $\mathcal{X} \otimes \mathcal{Y}$.

The space of all linear mappings between two Hilbert spaces $\mathcal{X}, \mathcal{Y}$ is denoted as $\mathrm{L}(\mathcal{X}, \mathcal{Y})$, and $\mathrm{L}(\mathcal{X})$ is $\mathrm{L}(\mathcal{X}, \mathcal{X})$ for short. An inner product is defined on the space $\mathrm{L}(\mathcal{X}, \mathcal{Y})$ as $\langle A, B\rangle = \mathrm{Tr}(A^*B)$. In quantum computation, we also call these linear mappings *operators*. An operator $X \in \mathrm{L}(\mathcal{X})$ is *unitary* if $X^*X = \mathbb{I}_{\mathcal{X}}$, and the subset of all unitary operators is denoted as $\mathrm{U}(\mathcal{X})$; *Hermitian* if $X^* = X$, denoted as $\mathrm{Herm}(\mathcal{X})$; *semidefinite* if X is Hermitian and all eigenvalues are nonnegative, denoted as $\mathrm{Pos}(\mathcal{X})$; *projection operator* if $XX^2 = X$, denoted as $\mathrm{Proj}(\mathcal{X})$; *density operator* if X is positive semidefinite and $\mathrm{Tr}(X) = 1$, denoted

as $\mathrm{D}(\mathcal{X})$. An operator $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$ is *isometry* if $A^*A = \mathbb{I}_{\mathcal{X}}$, and the subset of all isometry operators is denoted as $\mathrm{U}(\mathcal{X}, \mathcal{Y})$.

All Hermitian matrices only have real eigenvalues. If $n = \dim \mathcal{X}$, any $A \in \mathrm{Herm}(\mathcal{X})$ has n eigenvalues, and we usually represent them as $\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_n(A)$, with $\lambda_1(A)$ being the largest one and $\lambda_n(A)$ the smallest. Although all pure states can be expressed as column vectors in $\mathcal{X}$, density operators are actually a more general way to represent all possible, pure or mixed, quantum states. The density matrix corresponding to a pure state $|\psi\rangle$ is $|\psi\rangle\langle\psi|$. Otherwise it is a mixed state. If the system ρ is defined on a composite Hilbert space $\mathcal{X} \otimes \mathcal{Y}$, we can perform the *partial trace* over one of its subsystem, say $\mathcal{Y}$, as $\mathrm{Tr}_{\mathcal{Y}}(\rho) = \sum_y (\mathbb{I}_{\mathcal{X}} \otimes \langle y|)\rho(\mathbb{I}_{\mathcal{X}} \otimes |y\rangle)$, where $\{|y\rangle\}$ is an orthonormal basis on $\mathcal{Y}$. The purification of a mixed state $\rho \in \mathrm{D}(\mathcal{X})$, is a pure state $|\psi\rangle$ defined on space $\mathcal{X} \otimes \mathcal{Y}$, such that $\mathrm{Tr}_{\mathcal{Y}}(|\psi\rangle\langle\psi|) = \rho$, where $\mathcal{Y}$ is an auxiliary space.

Theorem 2.1 (Unitary equivalence of purification)**.** *Let $\rho \in \mathrm{D}(\mathcal{X})$ and suppose $|\psi\rangle, |\phi\rangle \in \mathcal{X} \otimes \mathcal{Y}$ satisfy*

$$\mathrm{Tr}_{\mathcal{Y}}(|\psi\rangle\langle\psi|) = \rho = \mathrm{Tr}_{\mathcal{Y}}(|\phi\rangle\langle\phi|). \tag{1}$$

There exists a unitary operator $U \in \mathrm{U}(\mathcal{Y})$ such that $|\phi\rangle = (\mathbb{I}_{\mathcal{X}} \otimes U)|\psi\rangle$.

An super-operator is a linear mapping $\Psi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$. The set of all super-operator is denoted as $\mathrm{T}(\mathcal{X}, \mathcal{Y})$. Due to the *Stinespring representation*, any super-operator $\Psi \in \mathrm{T}(\mathcal{X}, \mathcal{Y})$ can always be written as $\Psi(X) = \mathrm{Tr}_{\mathcal{Z}}(AXB^*)$ for all $X \in \mathrm{L}(\mathcal{X})$, where $\mathcal{Z}$ is an auxiliary space, and $A, B \in \mathrm{L}(\mathcal{X}, \mathcal{Y} \otimes \mathcal{Z})$. We define $\Psi^* \in \mathrm{T}(\mathcal{Y}, \mathcal{X})$ as the adjoint super-operator of Ψ if it satisfies that $\langle B, \Psi(A)\rangle = \langle \Psi^*(B), A\rangle$ for all operators $A \in \mathrm{L}(\mathcal{X}), B \in \mathrm{L}(\mathcal{Y})$.

A super-operator Ψ is *positive* if $\Psi(X) \in \mathrm{Pos}(\mathcal{Y})$ for any $X \in \mathrm{Pos}(\mathcal{X})$; Ψ *completely positive* if $\Psi \otimes \mathbb{I}_{\mathrm{L}(\mathcal{Z})}$ is positive for any choice of a complex vector space $\mathcal{Z}$; Ψ is *trace preserving* if $\mathrm{Tr}(\Psi(\mathcal{X})) = \mathrm{Tr}(\mathcal{X})$ for all $X \in \mathrm{L}(\mathcal{X})$. For a super-operator $\Psi \in \mathrm{T}(\mathcal{X}, \mathcal{Y})$ to be physically implementable (also called *admissible*), it must be completely positive and trace-preserving. Such super-operator is called a *channel*, represented as $\Psi \in \mathrm{C}(\mathcal{X}, \mathcal{Y})$. Using the Stinespring representation, $\Psi \in \mathrm{C}(\mathcal{X}, \mathcal{Y})$ if and only if $A = B \in \mathrm{U}(\mathcal{X}, \mathcal{Y})$. All channels represent the discrete-time changes in quantum systems. The difference between two channels is a super-operator.

Sometimes we want to measure the distance between different quantum states. The *trace distance* is $\frac{1}{2}\|\rho - \sigma\|_1$, where $\|\cdot\|_1$ is the *trace norm*, defined as $\|X\|_1 = \mathrm{Tr}(\sqrt{XX^*})$. *Fidelity* is another useful distance measurement, defined as $\mathrm{F}(\rho, \sigma) = \mathrm{Tr}(\sqrt{\sqrt{\rho}\,\sigma\sqrt{\rho}}) = \|\sqrt{\rho}\sqrt{\sigma}\|_1$.

Theorem 2.2 (*Uhlmann's theorem*)**.**

$$\mathrm{F}(\rho, \sigma) = \max_{|\psi\rangle, |\phi\rangle} |\langle\psi|\phi\rangle| \tag{2}$$

where $|\psi\rangle, |\phi\rangle$ are the purifications of ρ, σ respectively.

Theorem 2.3 (*Fuchs-van de Graaf inequalities*)**.**

$$1 - \mathrm{F}(\rho, \sigma) \leq \frac{1}{2}\|\rho - \sigma\|_1 \leq \sqrt{1 - \mathrm{F}(\rho, \sigma)^2} \tag{3}$$

Fuchs-van de Graaf inequalities connects the trace distance and the fidelity. The trace distance is bounded by the fidelity.

2.2 Quantum Interactive Proof

A quantum interactive game is the analog of the quantum interactive proof system. We assume an all-powerful *prover*, who tries to convince a computationally bounded *verifier* that the input x is of certain property, through m-turn communication. We say such is a *m-turn interactive game*, or the verifier is a *m-turn verifier*, or the prover is a *m-turn prover*. The verifier want to protect himself from being conceived by the prover. The prover and verifier are both able to do quantum computation, assisted with their private memory registers untouchable by the other party, and they have a quantum channel to communicate. As in the classical case, the verifier is allowed to use private coins as the source of randomness.

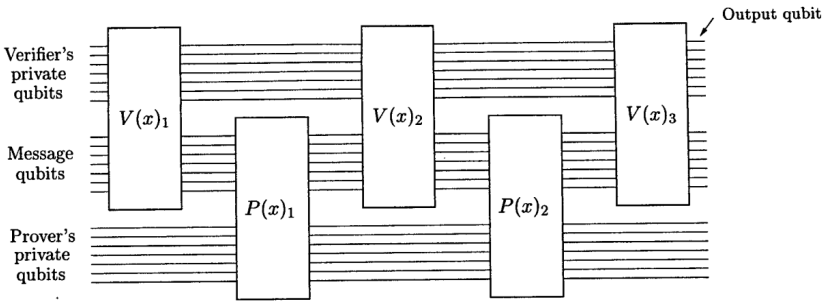

Figure 1: A four-turn interactive game.[1]

Let $n = \lfloor \frac{m}{2} \rfloor$, $V(x) = (V_1, V_2, \cdots, V_{n+1})$ is the action of the verifier; $P(x) = (P_0, \cdots, P_n)$ if m is odd, or $P(x) = (P_1, \cdots, P_n)$ if m is even, is the action of the prover. V, P are both decided by the input x. All V_i, P_i must be able to be described by quantum channels. By adding auxiliary qubits, we can always purify the circuit, which means all V_i, P_i are unitary or isometry operators, and the joint state of all of the co-existing registers at each instant is a pure state. Since the verifier is computationally limited, $V(x)$ should be able to be computed within polynomial time. Conventionally, originally we set all qubits to $|0\rangle$, then the registers are subject to $V(x), P(x)$. No matter m is odd or even, the prover will always send the last message to the verifier, and the first qubit of the result of P_{n+1} is the output qubit of the system, denoted as $\langle V, P\rangle(x)$. We will measure the output qubit with respect to the computational basis $\{|0\rangle, |1\rangle\}$, and accept if the measurement is 1, otherwise reject. An example is shown in the figure 2.2. It is a four-turn interactive game.

Once the protocol of an interactive game is set, and the input x is given, we can assume the behavior of the verifier is fixed. The prover should try his best to increase the probability that the verifier accepts, as long as his action is physically admissible. We define the *value* of a given verifier $V(x)$ as $\omega(V(x)) = \max_P \langle V, P\rangle(x)$, which is the maximal probability that a verifier could accept given the input x.

Definition 2.4. A promise problem is a pair $A = (A_{yes}, A_{no})$, where $A_{yes} \cap A_{no} = \varnothing, A_{yes} \cup A_{no} \subseteq \Sigma^*$. It is required that for any $x \in A_{yes}$, the output is 1; for any $x \in A_{no}$, the output is 0. If $x \notin A_{yes} \cup A_{no}$, the output can be anything.

Definition 2.5. A promise problem $A = (A_{yes}, A_{no})$ is contained in the complexity class $\mathrm{QIP}_{a,b}(m)$ if there exists a polynomial-time computable function V, that possesses the following properties:

1. For every string $x \in A_{yes} \cup A_{no}$, one has that $V(x)$ is an encoding of a quantum circuit description of an m-turn verifier in an interactive game.
2. *Completeness:* For every string $x \in A_{yes}$, it holds that $\omega(V(x)) \geq a$.
3. *Soundness:* For every string $x \in A_{no}$, it holds that $\omega(V(x)) \leq b$

When when say $\mathrm{QIP}(m)$ without specification, usually it refers to $\mathrm{QIP}_{\frac{2}{3},\frac{1}{3}}(m)$.

Theorem 2.6. *$QIP = QIP_{1,2^{-p}}(3)$, where p is a constant or a polynomial function of the input length.*

Theorem 2.6 is such a fascinating property of quantum interactive proof system. It means that we can *parallelize* the quantum interactive proof systems to a high degree. Any quantum interactive proof system can be completed using only three rounds of interaction. There is no similar theorem for classical interactive proof systems. Actually it would cause the polynomial hierarchy collapses, namely $\mathsf{PH} = \Sigma_2$, if similar result holds, which is conjectured to be not likely by many researcher.

The proof of theorem 2.6 contains three steps:

[1] Figure 1 in [4]

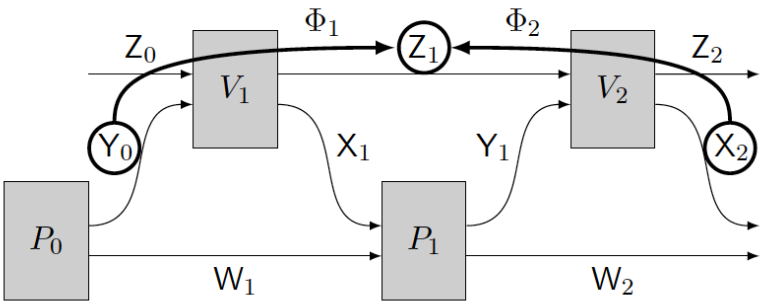

Figure 2: A purified three-turn interactive game. [2]

1. *Perfect completeness:* $\mathrm{QIP}_{a,b}(m) \subseteq \mathrm{QIP}_{1,1-(a-b)^2}(m+2)$;
2. *Parallelization:* $\mathrm{QIP}_{1,1-\varepsilon}(m) \subseteq \mathrm{QIP}_{1,1-\frac{\varepsilon^2}{4m^2}}(3), \forall m \in poly$;
3. *Amplification:* $\mathrm{QIP}_{1,b}(3) \subseteq \mathrm{QIP}_{1,b^{poly}}(3)$.

If you look into the proof carefully, you will find that the verifier does not really need a private random coin, a public coin works, too.

Corollary 2.7. *QIP = QMAM.*

The full proofs of theorem 2.6 and corollary 2.7 are beyond the scope of this lecture note. Curious readers can look at the article by Kitaev and Watrous[4] for more details.

3 A QIP-Complete Problem: *Close Images*

The *close images* problem is actually the first QIP-complete problem ever defined. Proposed in 2000 by Kitaev and Watrous[4], it is a problem closely related to the structure of quantum circuits.

Definition 3.1 (*close images*, $\mathrm{CI}_{\alpha,\beta}(\Phi_1, \Phi_2)$)**.** For constants $0 < \beta < \alpha \leq 1$, the input consists of two polynomial-time computable quantum circuits Φ_1, Φ_2 agreeing on the number of output qubits they produce. The promise problem is to distinguish the following two cases

Yes: $\mathrm{F}(\Phi_1(\rho_1), \Phi_2(\rho_2)) \geq \alpha$ for some choice of input states ρ_0, ρ_1;

No: $\mathrm{F}(\Phi_1(\rho_1), \Phi_2(\rho_2)) \leq \beta$ for all choices of input states ρ_0, ρ_1.

Corollary 3.2. $\mathrm{CI}_{1,\frac{1}{2}}$ *is QIP-complete.*

Proof. It follows directly from theorem 2.6. We will show that any $L \in \mathrm{QIP}_{1,\frac{1}{4}}(3)$ can be reduced to a $\mathrm{CI}_{1,\frac{1}{2}}$ problem. Figure 3 is an illustration of a three-turn interactive game. We assume the circuit is purified, thus V_1, P_1, V_2 are unitary, and the joint state of all co-existing registers at at moment is a pure state. We can always perform purification by adding auxiliary input qubits.

For now let's assume V, P are both fixed. We define two channels, $\Phi_1 \in \mathrm{C}(\mathcal{Y}_0, \mathcal{Z}_1), \Phi_2 \in \mathrm{C}(\mathcal{X}_2, \mathcal{Z}_1)$, as

$$\begin{aligned} \Phi_1(\rho_1) &= \mathrm{Tr}_{\mathcal{X}_1}(V_1(|0\cdots 0\rangle\langle 0\cdots 0| \otimes \rho_1)V_1^*) \\ \Phi_2(\rho_2) &= \mathrm{Tr}_{\mathcal{Y}_1}(V_2^*(|1\rangle\langle 1| \otimes \rho_2)V_2) \end{aligned} \tag{4}$$

[2] Figure 4.9 in [5]

for every $\rho_1 \in \mathrm{D}(\mathcal{Y}_0), \rho_2 \in \mathrm{D}(\mathcal{X}_2)$, as marked in figure 3. Let's assume $|0\cdots 0, \phi_1\rangle$ is the pure state of whole system corresponding to ρ_1, and $|1, \phi_2\rangle$ corresponds to ρ_2. For a certain message $\rho_1 \in \mathcal{Y}_0$ sent from the prover, the acceptance probability is

$$\begin{aligned} &\max_{|\phi_2\rangle \in \mathrm{D}(\mathcal{X}_2 \otimes \mathcal{W}_2)} |\langle 1, \phi_2|(V_2 \otimes \mathbb{I}_{\mathcal{W}_2})(\mathbb{I}_{\mathcal{Z}_1} \otimes P_1)(V_1 \otimes \mathbb{I}_{\mathcal{W}_1})|0\cdots 0, \phi_1\rangle|^2 \\ &= \max_{|\phi_2\rangle \in \mathrm{D}(\mathcal{X}_2 \otimes \mathcal{W}_2)} |\langle 1, \phi_2|(V_2 \otimes \mathbb{I}_{\mathcal{W}_2})(V_1 \otimes \mathbb{I}_{\mathcal{W}_1})|0\cdots 0, \phi_1\rangle|^2 \end{aligned} \tag{5}$$

We can eliminate P_1, because P_1 cannot affect the register Z_2, and we are maximizing in the space $|\phi_2\rangle \in \mathrm{D}\mathcal{X}_2 \otimes \mathcal{W}_2$. If the maximum of the left-hand side is reached when $|\phi_2\rangle = |\psi\rangle$, due to the theorem 2.1, we can always find another $|\hat{\psi}\rangle$, such that

$$(\mathbb{I}_{\mathcal{Z}_1} \otimes P_1^*)(V_2^* \otimes \mathbb{I}_{\mathcal{W}_2})|1, \psi\rangle = (V_2^* \otimes \mathbb{I}_{\mathcal{W}_2})|1, \hat{\psi}\rangle. \tag{6}$$

The maximum acceptance probability of V given P is

$$\omega(V)|_P = \max_{|\phi_1\rangle, |\phi_2\rangle} |(\langle 1, \phi_2|V_2)(V_1|0\cdots 0, \phi_1\rangle)|^2. \tag{7}$$

$V_2^*|1, \phi_2\rangle, V_1|0\cdots 0, \phi_1\rangle$ are purifications of $\Phi_1(\rho_1), \Phi_2(\rho_2)$ respectively. And since we are going to maximize over all possible prover, and the prover is all-powerful, we can find a corresponding P for all possible purification of $\Phi_1(\rho_1), \Phi_2(\rho_2)$. From the property of fidelity shown in theorem 2.2,

$$\omega(V) = \max_{\substack{\rho_1 \in \mathrm{D}(\mathcal{Y}_0) \\ \rho_2 \in \mathrm{D}(\mathcal{X}_2)}} F(\Phi_1(\rho_1), \Phi_2(\rho_2))^2. \tag{8}$$

Compare with the definition of CI. If $\omega(V) = 1$, there should exists $\rho_1 \in \mathcal{Y}_0, \rho_2 \in \mathcal{X}_2$, such that $F(\Phi_1(\rho_1), \Phi_2(\rho_2)) = 1$; if $\omega(V) \leq \frac{1}{4}$, $F(\Phi_1(\rho_1), \Phi_2(\rho_2)) \leq \frac{1}{2}, \forall\ \rho_1 \in \mathcal{Y}_0, \rho_2 \in \mathcal{X}_2$. □

Now all we need to do is to prove that $\mathrm{CI}_{1,\frac{1}{2}} \in$ PSPACE. We will show this in the following several sections.

4 CI as a Min-Max Problem

In this section, we are going to reduce CI to a min-max problem. This is the critical step in this proof. The min-max form allow us to use the matrix multiplicative weights update method coming next section to solve, which can be implemented as a PSPACE algorithm.

Lemma 4.1. *We are given* $\mathrm{CI}_{1,\frac{1}{2}(\Phi_1,\Phi_2)}$, *where* $\Phi_1 \in \mathrm{C}(\mathcal{X}_1, \mathcal{Y}), \Phi_2 \in \mathrm{C}(\mathcal{X}_2, \mathcal{Y})$. *Define* $\Psi_1 = \Phi_1 \otimes \mathrm{Tr}_{\mathcal{X}_2} \in \mathrm{C}(\mathcal{X}_1 \otimes \mathcal{X}_2, Y), \Psi_2 = \mathrm{Tr}_{\mathcal{X}_1} \otimes \Phi_2 \in \mathrm{C}(\mathcal{X}_1 \otimes \mathcal{X}_2, Y), \Xi = \Psi_1 - \Psi_2 \in \mathrm{T}(\mathcal{X}_1 \otimes \mathcal{X}_2, Y)$. *We define a real number value*

$$\eta = \min_{\rho \in \mathrm{D}(\mathcal{X}_1 \otimes \mathcal{X}_2)} \max_{\Pi \in \Gamma} \langle \Pi, \Xi(\rho) \rangle, \tag{9}$$

where $\Gamma = \{\Pi : \Pi \in \mathrm{L}(\mathcal{Y}), 0 \leq \Pi \leq \mathbb{I}_{\mathcal{Y}}\}$. *Then*

$$[\mathrm{CI}_{1,\frac{1}{2}}(\Phi_1, \Phi_2) = 1] \Rightarrow [\eta = 0];\ [\mathrm{CI}_{1,\frac{1}{2}}(\Phi_1, \Phi_2) \leq \frac{1}{2}] \Rightarrow [\eta \geq \frac{1}{2}]. \tag{10}$$

Proof. Since $\mathrm{D}(\mathcal{X}_1 \otimes \mathcal{X}_2), \mathrm{Proj}(\mathcal{Y})$ are convex and compact sets, and $\langle \Pi, \Xi(\rho) \rangle$ is a bilinear function over them. it follows from *von Neumann's Min-Max Theorem* that

$$\eta = \min_{\rho \in \mathrm{D}(\mathcal{X}_1 \otimes \mathcal{X}_2)} \max_{\Pi \in \Gamma} \langle \Pi, \Xi(\rho) \rangle = \max_{\Pi \in \Gamma} \min_{\rho \in \mathrm{D}(\mathcal{X}_1 \otimes \mathcal{X}_2)} \langle \Pi, \Xi(\rho) \rangle \tag{11}$$

We look at *yes* and *no* cases respectively:

Yes: If $\mathrm{CI}_{1,\frac{1}{2}}(\Phi_1,\Phi_2)=1$, it means that we can find ρ_1,ρ_2, such that $\mathrm{F}(\Phi_1(\rho_1),\Phi_2(\rho_2))=1\Rightarrow\Phi_1(\rho_1)=\Phi_2(\rho_2)$. Set $\rho=\rho_1\otimes\rho_2$, then $\Xi(\rho)=0,\eta=\max_\Pi\langle\Pi,0\rangle=0$.

No: If $\mathrm{CI}_{1,\frac{1}{2}}(\Phi_1,\Phi_2)=0$, $\mathrm{F}(\Phi_1(\rho_1),\Phi_2(\rho_2)\leq\frac{1}{2},\ \forall\rho\in\mathrm{D}(\mathcal{X}_1\otimes\mathcal{X}_2)$. Due to theorem 2.3

$$\|\Xi(\rho)\|_1=\|\Phi_1(\rho_1)-\Phi_2(\rho_2)\|_1\geq 2-2\mathrm{F}(\Phi_1(\rho_1),\Phi_2(\rho_2))\geq 1, \tag{12}$$

where $\rho_1=\mathrm{Tr}_{\mathcal{X}_2}(\rho),\rho_2=\mathrm{Tr}_{\mathcal{X}_1}(\rho)$.

$$\eta=\min_{\rho\in\mathrm{D}(\mathcal{X}_1\otimes\mathcal{X}_2)}\max_{\Pi\in\Gamma}\langle\Pi,\Xi(\rho)\rangle\geq\min_{\rho\in\mathrm{D}(\mathcal{X}_1\otimes\mathcal{X}_2)}\frac{1}{2}\|\Xi(\rho)\|_1\geq\frac{1}{2}. \tag{13}$$

□

We can see that there is a clear gap between *yes* and *no* instances. If we can approximately compute η, which is an equilibrium value, then we can distinguish these two cases. In the following section, we are going to introduce an algorithm to compute η.

5 The Matrix Multiplicative Weights Update Method

The *matrix multiplicative weights update method* is a meta-algorithm for convex optimization problems. We will use an algorithm in this family that can be used to compute the equilibrium value η defined in equation (11). They algorithm is as such:

Algorithm 1 (the matrix multiplicative weights update method)**.** It is an iterative algorithm:

1. Set $X_1=\mathbb{I}_N$ and $T=\lceil 2\frac{\ln(N)}{\varepsilon^2}\rceil$;
2. For each $t=1,2,\cdots,T$, let
$$\rho_t=\frac{X_t}{\mathrm{Tr}(X_t)}, \tag{14}$$
let $\Pi_t\in\mathrm{Proj}(\mathbb{C}^M)$ be the projection operator corresponding to the positive eigenspace of $\Xi(\rho_t)$, and let
$$X_{t+1}=\exp(-\varepsilon\Xi^*\sum_{i=1}^{t}\Pi_i) \tag{15}$$
3. Output
$$\hat{\eta}=\frac{1}{T}\sum_{t=1}^{T}\langle\Pi_t,\Xi(\rho_t)\rangle. \tag{16}$$

Let's see what this algorithm does. It iterates for T times. In each round, it normalizes X_t to get a density matrix ρ_t. Π_t being the projection operator corresponding to the positive eigenspace of $\Xi(\rho_t)$ satisfies

$$\Pi_t=\operatorname*{argmax}_{\Pi\in\Gamma}\langle\Pi,\Xi(\rho_t)\rangle. \tag{17}$$

and then update X_t using the equation (15). The final output in equation (16) is

$$\hat{\eta}=\operatorname*{avg}_{i=1}^{T}\max_{\Pi\in\Gamma}\langle\Pi,\Xi(\rho_i)\rangle \tag{18}$$

We will later prove that $\hat{\eta}$ is a good approximation of η. Before proving, we need to introduce several lemmas. Presumably ε is a small number, we assume $\varepsilon<\frac{1}{4}$ without loss of generality.

Lemma 5.1. *Let N be a positive integer, let $H \in \mathrm{Herm}(\mathbb{C}^N)$ be a Hermitian operator satisfying $\|H\| \leq 1$, and let $\rho \in \mathrm{D}(\mathbb{C}^N)$ be a density operator. For every positive real number $\varepsilon > 0$, it holds that*

$$\langle \rho, \exp(-\varepsilon H)\rangle \leq \exp(-\varepsilon \exp(-2\varepsilon)\langle \rho, H\rangle) \cdot \exp(2\varepsilon \sinh(2\varepsilon)) \tag{19}$$

Lemma 5.2. *Let T, N be positive integers, let $H_1, H_2, \cdots, H_T \in \mathrm{Herm}(\mathbb{C}^N)$ be Hermitian operators satisfying $\|H_t\| \leq 1$ for each $t \in \{1, 2, \cdots, T\}$, and let $\varepsilon > 0$. Define*

$$X_1 = \mathbb{I}, X_{t+1} = \exp(-\varepsilon \sum_{i=1}^{t} H_i), \rho_t = \frac{X_t}{\mathrm{Tr}(X_t)} \tag{20}$$

for each $t \in \{1, \cdots, T\}$. It holds that

$$\lambda_{min}(\sum_{t=1}^{T} H_t) \geq \exp(-2\varepsilon) \sum_{t=1}^{T} \langle \rho_t, H_t\rangle - \frac{\ln(N)}{\varepsilon} - 2T \sinh(2\varepsilon). \tag{21}$$

Proof. For each $t \in \{1, 2, \cdots, T\}$, one has

$$\mathrm{Tr}(X_t \exp(-\varepsilon H_t)) = \mathrm{Tr}(X_t)\langle \rho_t, \exp(-\varepsilon H_t)\rangle \tag{22}$$

by a matrix inequality know as the *Golden-Thompson inequality*, which states that $\mathrm{Tr}(\exp(A + B)) \leq \mathrm{Tr}(\exp(A)\exp(B))$ for every choice of Hermitian operators A and B.By applying this inequality repeatedly, and noting that $\mathrm{Tr}(X_1) = N$, one finds that

$$\mathrm{Tr}(X_{T+1}) \leq N \prod_{t=1}^{T} \langle \rho_t, \exp(-\varepsilon H_t)\rangle. \tag{23}$$

Because the trace of a positive semidefinite operator is at least as large as its largest eigenvalue, it follows that

$$\mathrm{Tr}(X_{T+1}) \geq \lambda_{max}(X_{T+1}) = \exp(-\varepsilon \lambda_{min}(\sum_{t=1}^{T} H_t)). \tag{24}$$

Combining equation (23) and equation (24), we can get

$$\lambda_{min}(\sum_{t=1}^{T} H_t) \geq -\frac{\ln(N)}{\varepsilon} - \sum_{t=1}^{T} \frac{\ln\langle \rho_t, \exp(-\varepsilon H_t)\rangle}{\varepsilon}. \tag{25}$$

Combining equation (25) with equation (19), we can get equation (21), which is what we want to prove. □

Theorem 5.3.

$$\eta \leq \frac{1}{T} \sum_{t=1}^{T} \langle \Pi_t, \Xi(\rho_t)\rangle \leq \eta + 16\varepsilon \tag{26}$$

Proof. From the equation (18), it is easy to see that $\eta \leq \hat{\eta}$, since $\eta \leq \max_{\Pi \in \Gamma} \langle \Pi, \Xi(\rho_i)\rangle, \forall\ i$.

The upper bound is deduced from lemma 5.2. Since Ξ is a difference between two channels, thus $\|\Xi\| \leq 1$. Set $H_t = \Xi^*(\Pi_t)$ for $t \in \{1, 2, \cdot, T\}$, we get

$$\frac{1}{T} \sum_{t=1}^{T} \langle \Pi_t, \Xi(\rho_t)\rangle \leq \exp(2\varepsilon)\lambda_{min}\left(\Xi^*\left(\frac{1}{T}\sum_{t=1}^{T} \Pi_t\right)\right) + \frac{\ln(N)\exp(2\varepsilon)}{\varepsilon T} + (\exp(4\varepsilon) - 1). \tag{27}$$

Notice that

$$\lambda_{min}\left(\Xi^*\left(\frac{1}{T}\sum_{t=1}^{T} \Pi_t\right)\right) = \min_{\rho} \left\langle \Xi(\rho), \frac{1}{T}\sum_{t=1}^{T} \Pi_t \right\rangle \leq \eta, \tag{28}$$

therefore

$$\frac{1}{T}\sum_{t=1}^{T}\langle \Pi_t, \Xi(\rho_t)\rangle \leq \exp(2\varepsilon)\eta + \frac{\varepsilon\exp(2\varepsilon)}{2} + (\exp(4\varepsilon)-1). \tag{29}$$

For any choice of $\delta \leq 1$, it holds that $\exp(\delta) - 1 \leq 2\delta$, and by combining this bound with the observation that $\eta \leq 1$, one obtains

$$\hat{\eta} = \frac{1}{T}\sum_{t=1}^{T}\langle \Pi_t, \Xi(\rho_t)\rangle \leq \eta + 16\varepsilon. \tag{30}$$

which completes the proof. □

If we set $\varepsilon = \frac{1}{128}$, it is sufficient to distinguish A_{yes} and A_{no}: the gap is $\frac{1}{2}$, while the error is $16\varepsilon = \frac{1}{8}$.

6 CI $\in$ PSAPCE

Now that we have a algorithm to compute η accurately enough to distinguish A_{yes} and A_{no}, the last step is whether we can implement this algorithm using only polynomial space. While it is not an easy step, because to design a space-efficient implementation is not very intuitive. However, thanks to Borodin's work[3] proving that PSPACE = NC(*poly*), we can seek to implement using bounded-depth boolean circuits instead.

NC(*poly*) is the class of promise problems computed by the polynomial-space uniform boolean circuits with polynomial depth. It can be seen as an extension of NC, which is the class of promise problems computed by logarithmic-space uniform boolean circuits with poly-logarithmic depth.

Fact 1. *If $f \in$ NC(poly), $g \in$ NC, then $g \circ f \in$ NC(poly).*

Fact 2. *Elementary matrix computations can be performed in NC. Matrix exponentials and positive eigenspace projections can be approximated to high precision in NC.*

Theorem 6.1. $\mathrm{CI}_{1,\frac{1}{2}}(\Pi_1, \Pi_2) \in$ *PSPACE.*

Sketch Proof. We have shown how to convert $\mathrm{CI}_{1,\frac{1}{2}}(\Pi_1, \Pi_2)$ as an equilibrium problem, then solve using algorithm 1. To decide $\mathrm{CI}_{1,\frac{1}{2}}(\Pi_1, \Pi_2)$, it suffices to compose the following families of boolean circuits.

1. A family of boolean circuits that output the representation of the quantum channel Ξ generated from the input x, namely, the descriptions of two mixed quantum circuits.
2. Follow algorithm 1. Compose all the operations in each iteration. Consider the fact that fundamental matrix operations can be done in NC and the number of iterations $T = \lceil \frac{2\ln N}{\varepsilon^2} \rceil$ is polynomial in the size of x since ε is a constant and N is exponential in the size of x.
3. The circuits to distinguish between the two promises by making use of the value returned in the circuits above.

The first family is easily done in NC(*poly*), by computing the product of a polynomial number of exponential-size matrices, which corresponds to the mixed quantum circuits. The second family is in NC by composing polynomial number of NC circuits. The third one is obviously in NC. The whole process is in NC(*poly*) by composing the NC(*poly*) and NC circuits above, and thus in PSPACE. □

Careful reader may have noticed that in the proof to theorem 6.1, the circuits we design can only *approximately* compute matrix exponentials and positive eigenspace projections in the looping step of algorithm 1. Will it cause any problem? The answer is no. We can ensure the approximation is within constant error δ. If we go through the proof again, adding small noise to ρ_t, Π_t, we can find that all the steps are basically unchanged except for additional error $poly(\delta)$. We will get a similar result, with the maximal error of $\hat{\eta}$ slightly increased. If we make δ small enough, the gap between A_{yes} and A_{no} will still be sufficient to distinguish. You may find more details on this issue in [1].

Hereby we finished the whole proof. We will conclude with the following corollary.

Corollary 6.2. *QIP = PSPACE.*

7 Discussion

QIP = PSPACE is a fascinating result, while at the same time a disappointing one. Since IP = PSPACE, it seems that quantum does not work in the interactive proof system. But we can also take an alternative view: this result shows the extraodinary expressive power of the interactive proof system, which overshadow the benefit of quantum computation. If we look a little close, we will find that even in the interactive proof system, quantum computing makes difference. As we see in theorem 2.6, the quantum interactive proof system can be parallelized to a constant number of turns without losing their expressive power, which is not known (and perhaps not expected) to be true for classical systems.

References

[1] Rahul Jain, Zhengfeng Ji, Sarvagya Upadhyay, and John Watrous. Qip= pspace. *Journal of the ACM (JACM)*, 58(6):30, 2011.

[2] Xiaodi Wu. Equilibrium value method for the proof of qip= pspace. *arXiv preprint arXiv:1004.0264*, 2010.

[3] Allan Borodin. On relating time and space to size and depth. *SIAM journal on computing*, 6(4):733–744, 1977.

[4] Alexei Kitaev and John Watrous. Parallelization, amplification, and exponential time simulation of quantum interactive proof systems. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 608–617. ACM, 2000.

[5] Thomas Vidick, John Watrous, et al. Quantum proofs. *Foundations and Trends® in Theoretical Computer Science*, 11(1-2):1–215, 2016.